IFC-Bank of Italy Workshop on "Machine learning in central banking"

19-22 October 2021, Rome / virtual event

Deep learning solutions for dynamic stochastic general equilibrium models[1]

Mo Ashtari and Vladimir Skavysh,
Bank of Canada

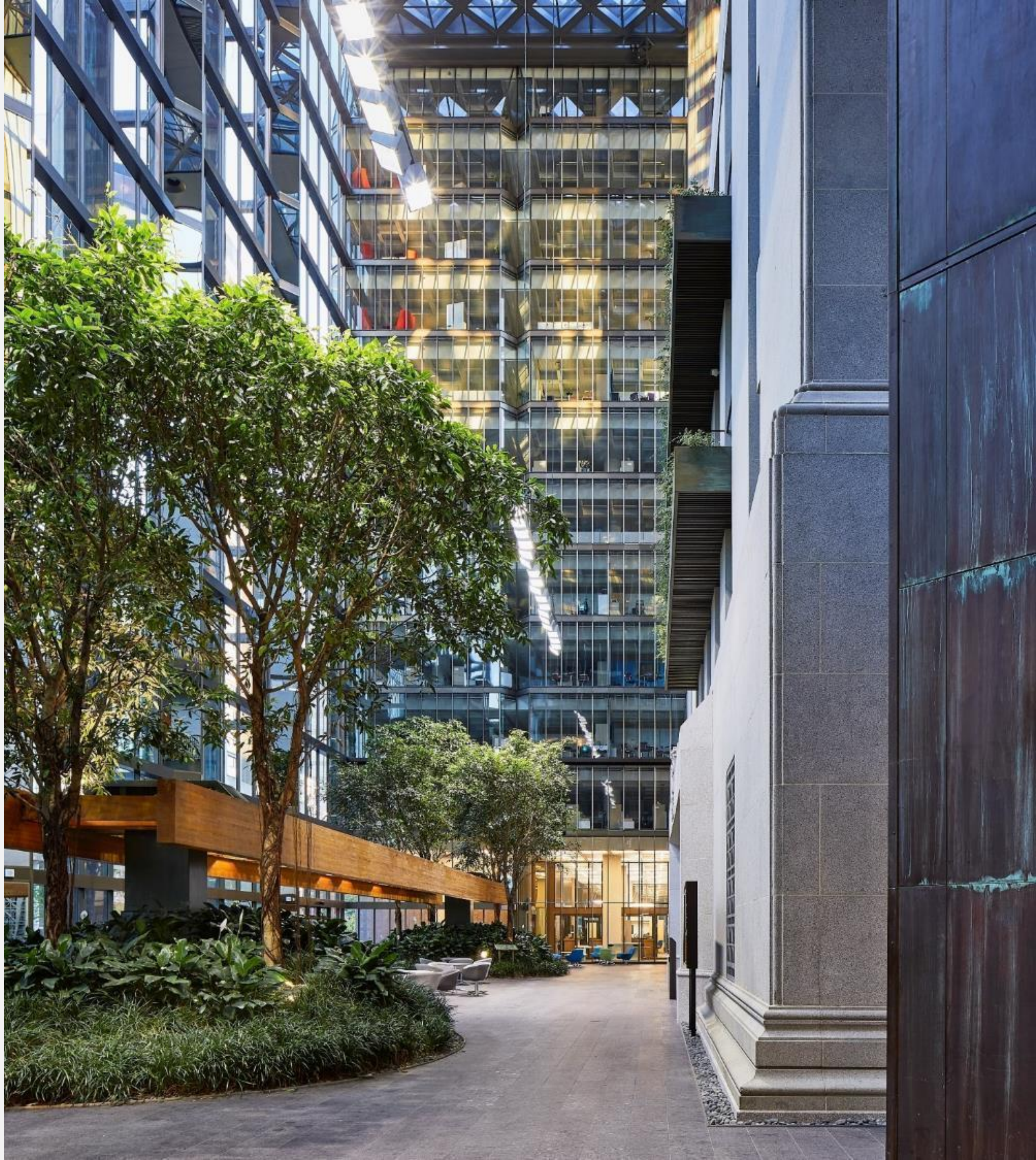
[1] This presentation was prepared for the Workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the central banks and other institutions represented at the event.

BANK OF CANADA
BANQUE DU CANADA

OCTOBER 2021

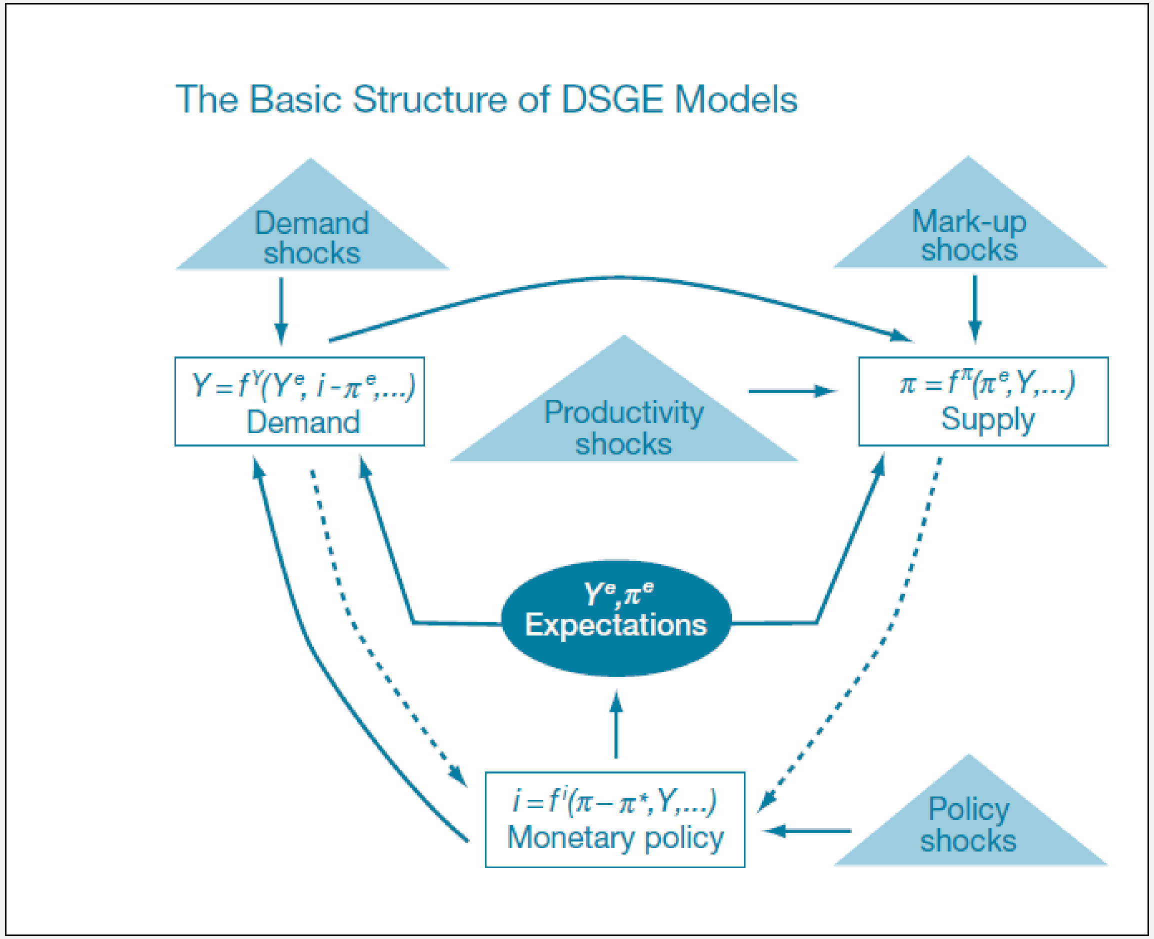
Deep Learning Solutions for Dynamic Stochastic General Equilibrium Models

Mo Ashtari, Vladimir Skavysh



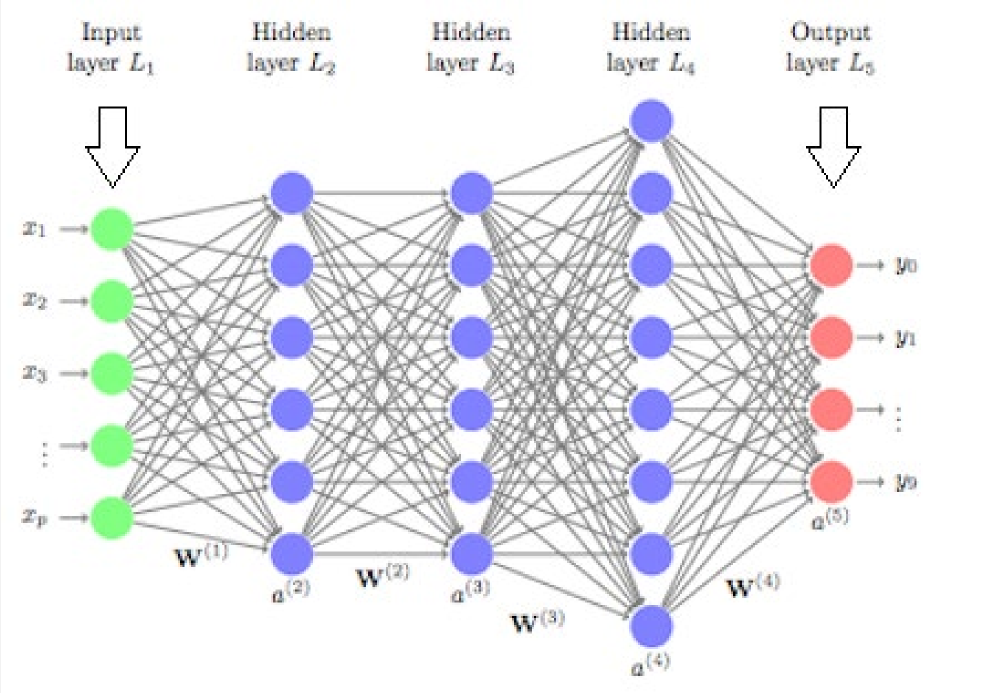
DSGE Models

- ❑ Dynamic: there are intertemporal problems and agents rationally form expectations
- ❑ Stochastic: exogenous stochastic processes may shift aggregates
- ❑ General Equilibrium: all markets are in equilibrium, although unpredictable shocks disturb this equilibrium for a while



Deep Learning

- ❑Machine Learning has shown to be a promising solutions in many fields as well as economics.
- ❑ Universal Approximation Theorem: A neural network with at least one hidden layer can approximate any Borel measurable function mapping finite-dimensional spaces to any desired degree of accuracy. (Hornik, Stinchcombe, and White, 1989)
- ❑ Breaking the curse of dimensionality: A one-layer neural network achieves integrated square errors of order $O(1/M)$, where M is the number of nodes. In comparison, for series approximations, the integrated square error is of order $O(1/(M^{2/N}))$ where N is the dimensions of the function to be approximated. (Barron 1993)
- ❑ Here, we used deep learning methods to solve DSGE models, starting with the simple Neoclassical growth model
- ❑ When applied to DSGE models, neural networks (NN) offer the following advantages:
  - ❑ Ability to solve high dimensional problems without the curse of dimensionality
  - ❑ High approximation power outside of the steady state



The Neo-Classical Growth Model

- In an economy with a representative household

$$\max_{c_t, k_{t+1}} E\left(\sum_{t=0}^{\infty} \beta^t u(c_t)\right)$$

where c_t is the consumption and $\mathrm{k_t}$ is the capital at time t, $E$is the expectation operator, β is the discount factor, and u is the utility function. The value function is defined by the Bellman operator:

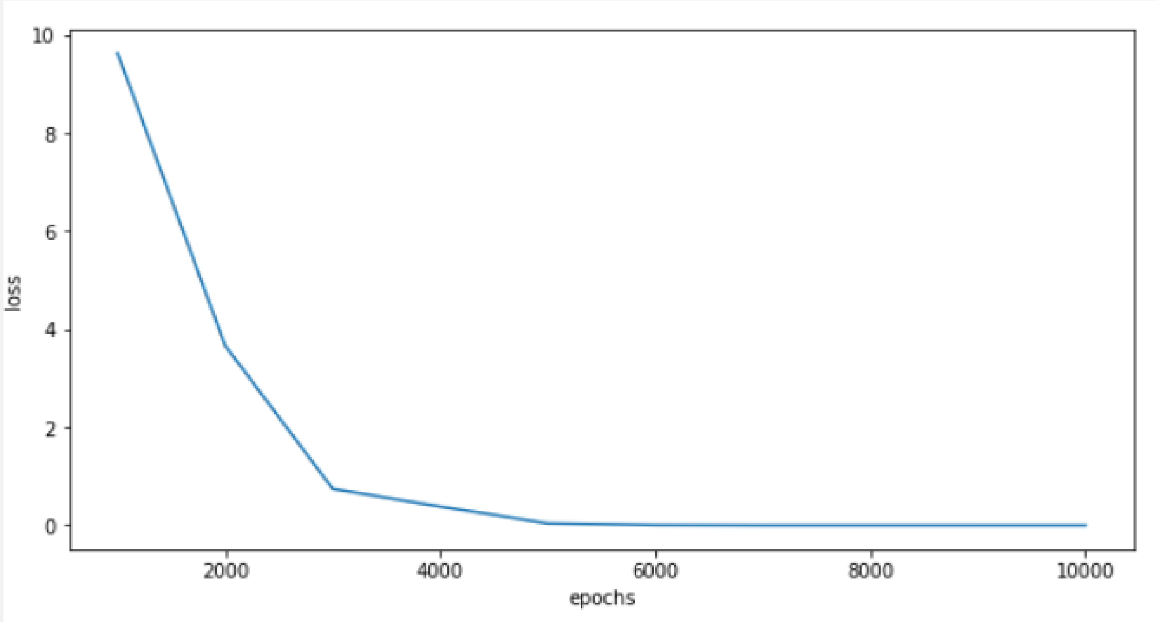
$$V(k_t, z_t) - \max_{k_{t+1}}[u(e^{z_t}k_t^{\alpha} + (1-\delta)k_t \ - k_{t+1}) + \ \beta V(k_{t+1}, z_{t+1})] = 0$$

where δ is the depreciation rate, and k and z are two state variables of the economy, capital and productivity

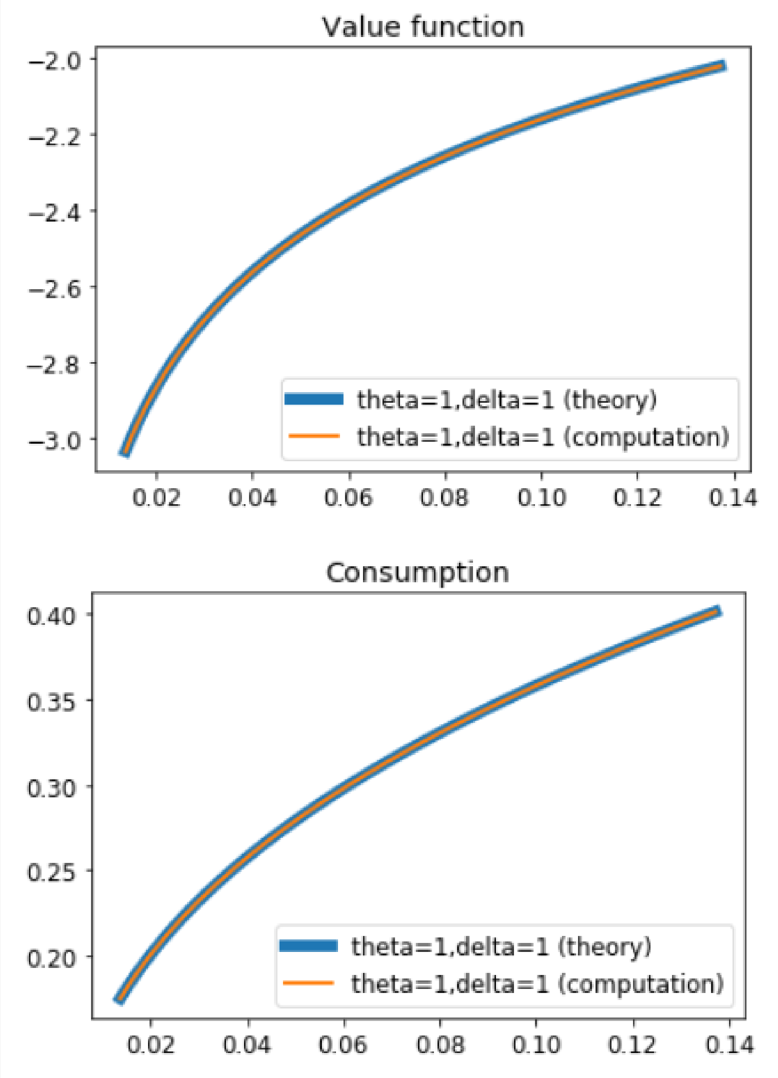
- We use deep learning to solve for the value function using the following steps:
  - Initialize V with a neural network parametrized by $\{\theta\}$
  - Make a random draw of k and z, as well as of future shocks $z_{t+1,1}$, and $z_{t+1,2}$
  - Calculate $\frac{\partial V(k,z)}{\partial k}$ and then solve for c
  - Solve for k_{t+1}
  - Compute the maximized Bellman equation
  - Compute the Bellman error
  - Update $\{\theta\}$ to minimize the error term. If it is smaller than a certain threshold stop. Otherwise loop back to make new random draws.

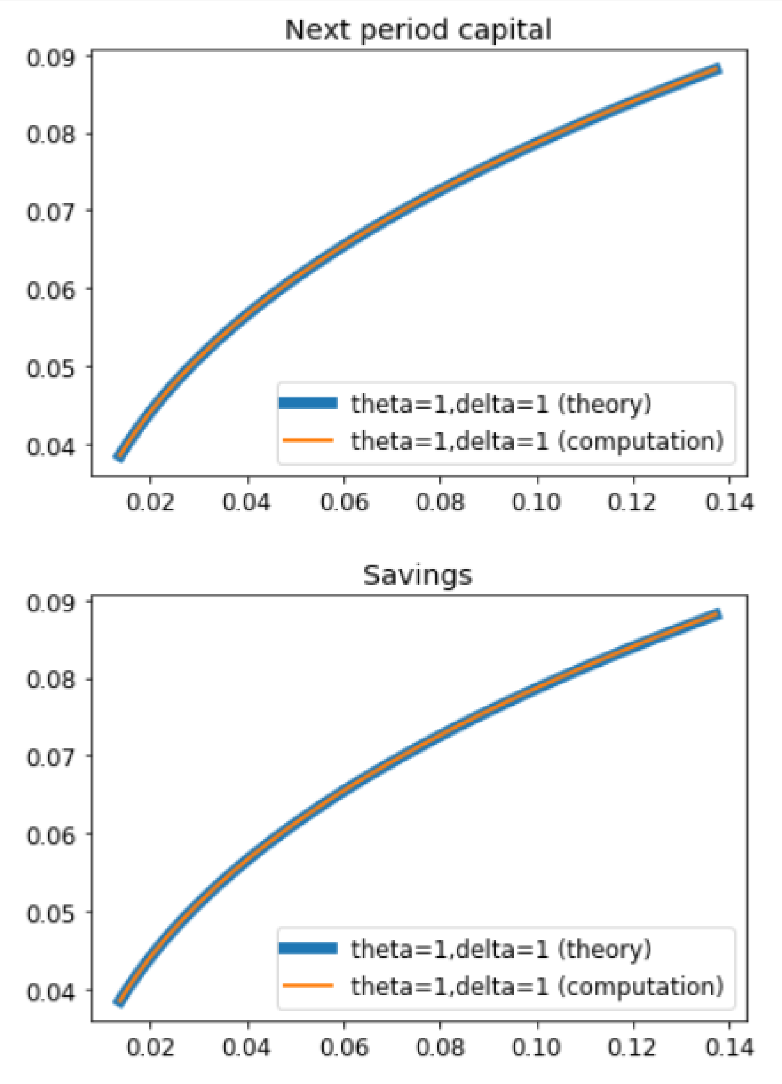
The Deep Learning Model

- We used both PyTorch and Tensorflow to tackle this problem.
- The model was tested with and without random market shocks
- k and z values were randomly chosen from a uniform and log-normal distributions, respectively
- In each epoch, the derivative of the network with respect to k is calculated and used to update the value of c
- A custom loss is defined and calculated using $||V(k,z) - T[V(k,z)]||$ and the network is minimized for loss using all trainable-weights
- The learning rate was set to 0.001 and is reduced step-wise after certain number of epochs



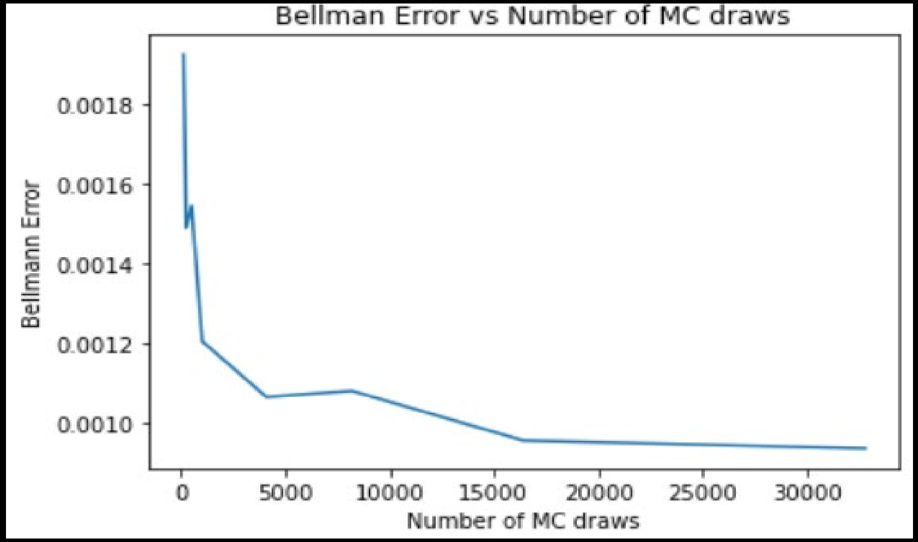
The Deep Learning Model





Limitation of the Deep Learning Framework for DSGE

- Monte Carlo is essential
  - provides unbiased estimator of the stochastic gradient with respect to random variables
  - possible to simultaneously estimate the decision function and to integrate with respect to future economic shocks
- Downside:
  - Low square-root rate of convergence



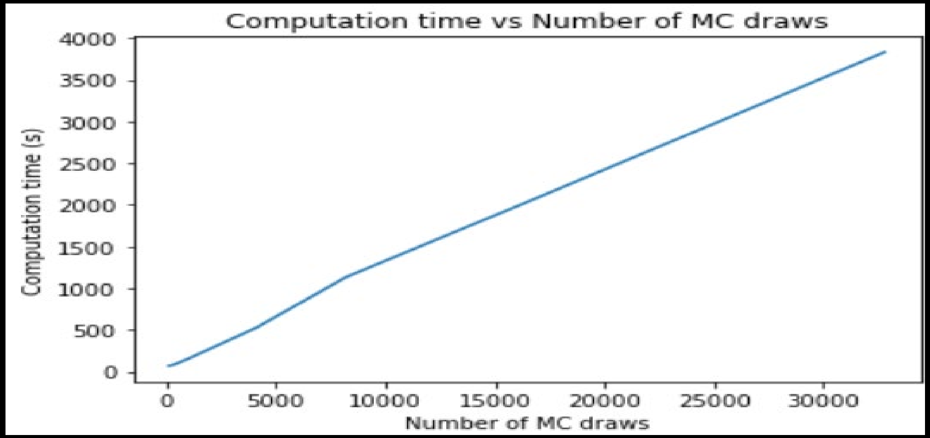


Figure: Code from Maliar, Lilia & Maliar, Serguei & Winant, Pablo, 2019. "Will Artificial Intelligence Replace Computational Economists Any Time Soon?," CEPR Discussion Papers 14024, C.E.P.R. Discussion Papers, Different numbers of MC draws were submitted to obtain Bellman Error

Classical vs Quantum Monte Carlo

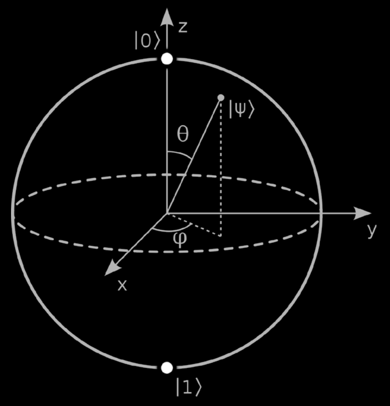
Classical (samples): $N \propto \frac{1}{\varepsilon^2}$

Quantum (gates): $N = \mathcal{O}\left(\frac{1}{\varepsilon}\right)$

Classical	Quantum
X samples	X gate applications
⬇	⬇
0.1 error	0.1 error
100X samples	10X gate applications
⬇	⬇
0.01 error	0.01 error

How Quantum Works

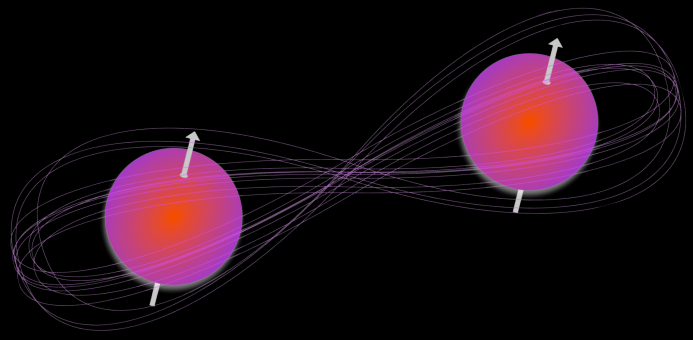
Superposition



Qubits can be a combination of 0 and 1 due to superposition, whereas binary bits can only represent 0 or 1

+

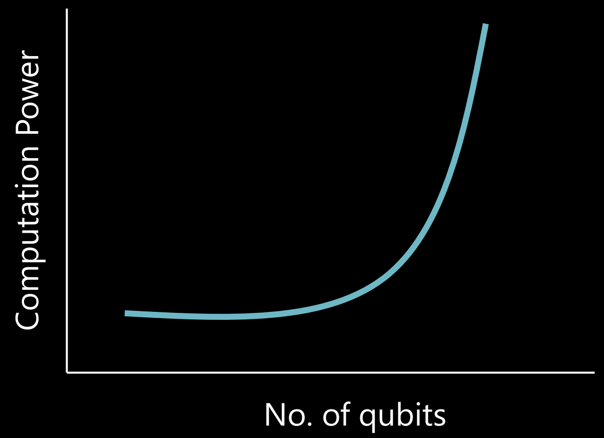
Entanglement



Quantum entanglement enables qubits to have special correlations between states, enabling greater information density

=

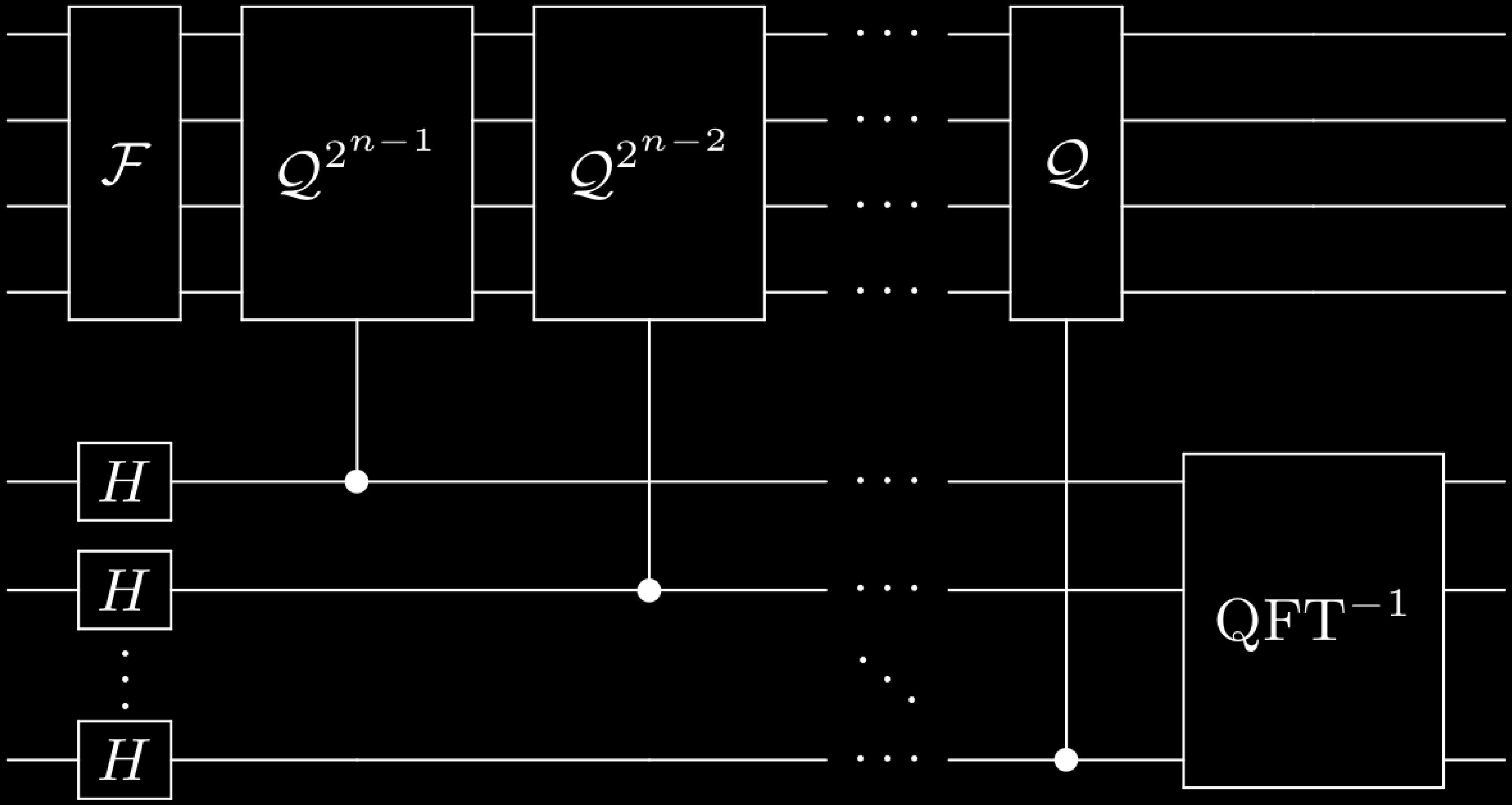
Faster computing



The power of quantum computers grows exponentially as individual qubits are added to the system

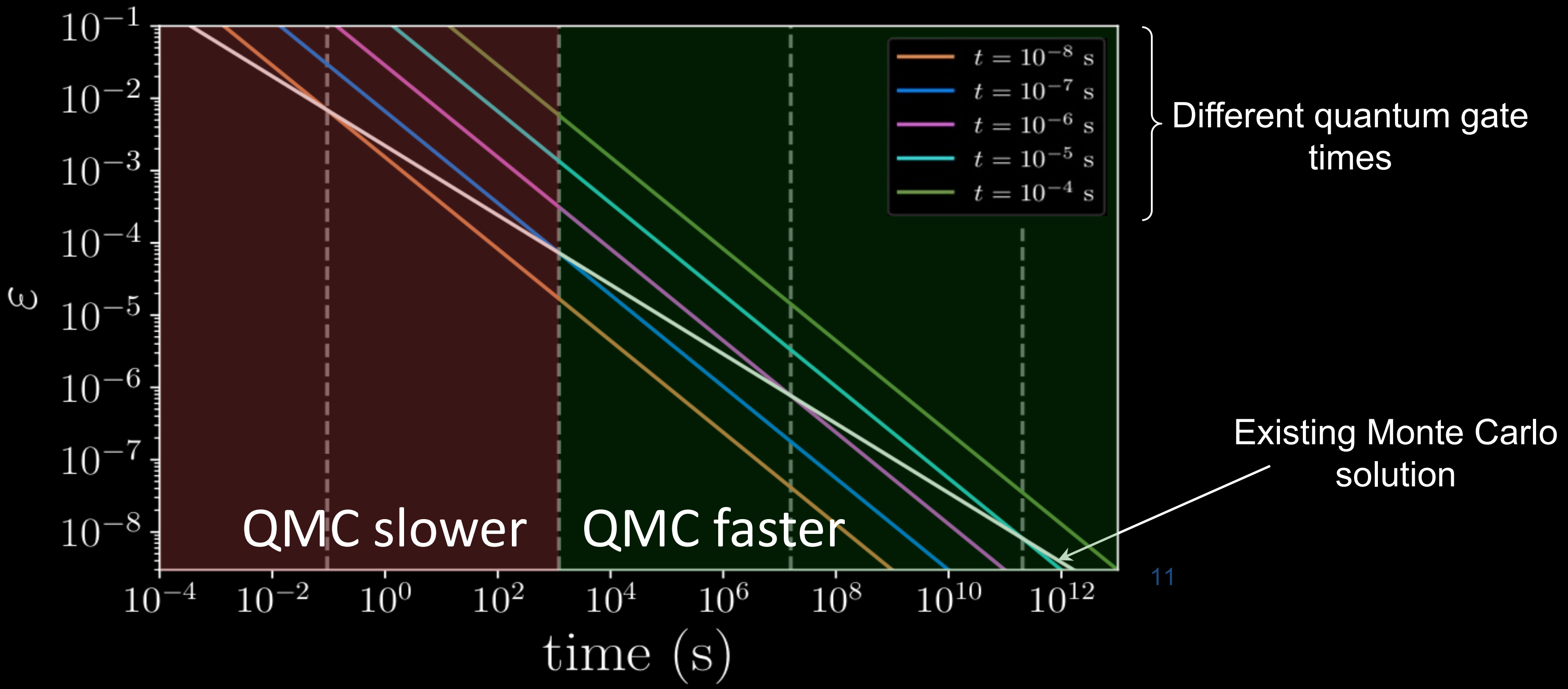
Figures curtesy of Xanadu Quantum Technologies, Inc.

Quantum Algorithm for Monte Carlo Estimation



Results for the Neoclassical Model

<u>The good news</u>: Assuming 10^{-7} s gate times, QMC is faster for simulations above 20 minutes



<u>The bad news</u>: requires circuit depth of 10,000+ (perhaps 5-15 years into the future)

Figure from Guala D., Skavysh V., Priazhkina S., Bromley T., 2021. "Economic Applications of Monte Carlo on a Quantum Computer"